

## REMARKS

Applicants respectfully request that the above-identified application be re-examined.

The Office Action mailed on June 24, 2005 ("Office Action"), rejected Claims 1-34. Claims 1-34 were rejected under 35 U.S.C. § 101, based on the assertion that "the language of claims 1-34 raises a question as to whether the claim is directed merely to an abstract idea that is not tied to a technological art, environment or machine, which would result in a practical application producing a useful, concrete, and tangible result to form the basis of statutory subject matter under 35 U.S.C. § 101." Office Action, page 2, item 4. More specifically, the Office Action rejected Claims 1-9 and 11-14 because they appear "to be comprised of software alone...." Office Action, page 2, item 5. The Office Action also rejected Claims 10 and 15-34 because these claims "are not limited to tangible embodiments." Additionally, the Office Action rejected Claims 16-34 because they "appear to be a data structure per se that is nonfunctional." Office Action, page 3, items 6 and 8.

Claims 1, 3, 5-8, 10, 11, 14-16, 18-28, and 30-33 were rejected under 35 U.S.C. § 102(e) as being fully anticipated by the teachings of U.S. Patent No. 6,698,012 B1, issued to Kossatchev et al. ("Kossatchev").

Claims 2, 4, 9, 12, 13, 17, 29, and 34 were rejected under 35 U.S.C. § 103(a) as being unpatentable in view of the teachings of Kossatchev taken in view of the teachings of U.S. Patent No. 6,754,850 B2, issued to Grey et al. ("Grey").

Claims 1-16 and 23 have been amended to further clarify the claim language.

Prior to discussing in detail why applicants believe that all of the claims in this application are allowable, a brief description of applicants' invention and a brief description of the teachings of the cited and applied references are provided. The following discussions of applicants' invention and the cited and applied references are not provided to define the scope or

LAW OFFICES OF  
CHRISTENSEN O'CONNOR JOHNSON KINDNESS<sup>PLLC</sup>  
1420 Fifth Avenue  
Suite 2800  
Seattle, Washington 98101  
206.682.8100

interpretation of any of the claims of this application. Instead, these discussions are provided to help the United States Patent and Trademark Office better appreciate important claim distinctions discussed thereafter.

### Applicants' Invention

The present invention is generally directed towards a software application's relationship to the application's run-time environment. More specifically, the invention is directed towards a framework and an engine for configuring and running applications in a run-time environment. Aspects of the framework and of the engine may be changed to suit a changing environment without changing the application. In one exemplary embodiment, the application may be a software test suite. The invention allows the test suite to be run as a single-threaded or a multi-threaded application and with individual tests within the suite running consecutively or concurrently without altering the underlying tests. The framework and engine are parameterized to accommodate different testing goals, such as regression testing and stress testing, without changing the test suite.

The invention creates data structures and functions to form a consistent framework around the application, connecting the application to the application's run-time environment. The invention provides common support and management features to different applications that would otherwise have to be built into each application. These features enable an application, written once, to run without modification in different runtime environments. FIGURE 2 shows how pieces of an exemplary embodiment of the invention relate to each other. The engine 200 uses the framework data structures 202 as input to run the application 204. The application may be one executable program or a collection of related, executable sub-applications 206. For

example, the application may be a suite of tests developed to exercise one feature of the computing environment 100 or one application under test.

The framework data structures 202 consist of two major parts, the application table 208 and the parameter table 210. The application table 208 contains information about each sub-application and includes special functions as part of the framework. The parameter table 210 holds variables used by the engine 200 and by the sub-applications 206 when the sub-applications 206 are executed. These two tables are described in more detail in FIGURE 4. In addition to the application table 208 and parameter table 210, the framework data structures include references to other framework functions. FIGURES 3a and 3b depict some of the framework functions, such as the module check function 300, which checks the availability of resources needed to run the application. Other examples of framework functions are module initialize function 302 and module terminate function 324. The parameter table 210 contains one entry 418 for each parameter that can be set and passed when the application 204 is run. Some of the parameters are defined by the developer of the application and other parameters are standard for all applications. Each parameter table entry contains a parameter name 420 and a description 422. The application table 208 contains an entry 432 for each sub-application 206. Each entry 432 includes fields for identifying the sub-application by name 434, description 436, and category 438. Additionally, among other fields, each entry 432 contains a flags field 440, which specifies run-time characteristics.

In summary, this invention is directed towards a framework and an engine for configuring and running applications in various run-time environments without changing the applications.

Kossatchev et al., U.S. Patent No. 6,698,012 B1 ("Kossatchev")

Kossatchev is generally directed towards the testing of software procedures and more specifically towards specifying the "behavior of a group of parallel procedures to be tested in a parallel mode separately from consecutive procedures." Kossatchev, Col. 1, lines 44-46. Kossatchev uses different kinds of application programming interface ("API") entities, such as procedures, operations, functions, methods, and subroutines, synonymously. Kossatchev, Col. 2, lines 44-47. Kossatchev discloses a method for testing of procedures comprising two stages. In the first stage, implementation-independent programs are generated. In the second stage, the implementation-independent programs are compiled into executable programs for a particular system under test. Kossatchev, Col. 3, lines 39-44. In Figure 1, Kossatchev discloses a verification system generator 1 to which the method for testing of procedures is applied. "The verification system 2 is generated for verifying a procedure interface 4 of a system under test (SUT) 3." Kossatchev, Col. 3, lines 1-5. In Figure 2, Kossatchev discloses a means 12 for generating specifications of the procedure interface 4. A test source generator 14 generates test source code based on the specifications. The specifications and test sources are stored in a repository 16. Kossatchev, Col. 3, lines 6-13. In Figure 1, Kossatchev discloses a test case parameter generator 32 that is used to generate constant arrays and programs that generate and select needed test case parameters. Kossatchev, Col. 4, lines 11-15. This is in contrast to the illustrated and described exemplary embodiment of the present invention where the application table 208 is used to hold information about sub-applications 206, and parameter table 210 is used to hold parameters for each sub-application 206 that are specified by the developer. Kossatchev also discloses test driver generator 30 used to generate test drivers. The test drivers execute tests on the SUT 3 using the test case parameters. Kossatchev, Col. 4, lines 16-21. The test drivers provide functions to initialize the procedure interface 4, prepare input values, call

tested procedures with test case parameters, and receive test results. Kossatchev, Col. 4, lines 31-36.

In summary, Kossatchev discloses a method for specifying the behavior of procedures and testing of procedures, which is independent of the implementation language of the system under test. This is in contrast to the present invention where a framework and an engine are used to configure and run applications.

Grey et al., U.S. Patent No. 6,754,850 B2 ("Grey")

Grey is directed towards a method for creating a program to be executed by a plurality of threads using batch synchronization section technique for execution synchronization. The batch synchronization section is a technique whereby a plurality of threads may be associated with one another as a batch of threads. Grey, Col. 3, lines 50-52. The batch synchronization section specifies a portion of the computer program for which the execution of the portion by the plurality of threads is to be synchronized. The batch synchronization section may specify an entry point and an exit point, which together define the portion of the program. Grey, Col. 3, lines 54-60. Grey discloses a method whereby each thread in the plurality of threads may execute the program until the thread arrives at the enter point for the batch synchronization section. The thread is blocked upon arriving at the enter point until all other threads arrive at the enter point. Once all threads have arrived at the enter point, execution of the program within the batch synchronization section proceeds according to an order of thread execution and a type specified for the batch synchronization section. For example, the type may be parallel, serial, or one-thread only. Grey, Col. 5, lines 52-58. Grey discloses a method that is used during the development of a program using an application development environment. During the development of the application, the developer may request a graphical user interface ("GUI")

useable to specify a batch synchronization section. The GUI may be displayed by the application development environment in a new window or dialog box. Grey, Col. 12, lines 21-29. This is in contrast to the present invention where a framework data structure 202 and an engine 200 are used to run an existing application in a run-time environment.

In summary, Grey discloses a method for creating a program in a development environment using a batch synchronization section technique for synchronizing multiple threads of execution. In contrast, the present invention is directed to a framework and an engine used to configure and run applications in a run-time environment.

#### Rejection of Claims 1-34 under 35 U.S.C. § 101

As noted above, Claims 1-34 were rejected under 35 U.S.C. § 101 by the Office Action for "raising a question as to whether the claims are directed merely to an abstract idea." Each specific ground of rejection raised by the Office Action under 35 U.S.C. § 101 is addressed accordingly as follows.

#### Claims 1-9 and 11-14

Claims 1-9 and 11-14 have been amended to further clarify the claim language and emphasize that the subject matter recited by these claims is statutory subject matter under 35 U.S.C. § 101. The Office Action rejected the above-mentioned claims for "appearing to be comprised of software alone without claiming ... utility." (Emphasis original). Applicants respectfully disagree. Amended independent Claim 1 recites, in its entirety:

1. A computer-implemented method for using a framework module to run an application, the framework module comprising an application table and a parameter table, the application table comprising one or more application table entries, the parameter table comprising one or more parameter table entries, the method comprising:

selecting an application table entry; and

LAW OFFICES OF  
CHRISTENSEN O'CONNOR JOHNSON KINDNESS<sup>PLLC</sup>  
1420 Fifth Avenue  
Suite 2800  
Seattle, Washington 98101  
206.682.8100

processing the selected application table entry, the processing comprising:

running a global initialize function referenced by the selected application table entry, running a sub-application referenced by the selected application table entry with one or more parameters referenced by one or more parameter table entries, and running a global terminate function referenced by the selected application table entry. (Emphasis added.)

Claims 1-9 and 11-14 have been amended to clarify that the recited method is "computer-implemented." It is well-known in the computer industry that software runs on a hardware platform. The concept of "running" directly indicates that there is a hardware platform on which to run. Therefore, the use of hardware in Claim 1 is implied by the phrase "run an application." Additionally, asserted utility (i.e. transformation of data) is clearly indicated by the claim language reciting "processing the selected application table entry." The term "processing" is defined as "[t]he manipulation of data within a computer system. Processing is the vital step between receiving data (input) and producing results (output)." (Emphasis added.) Peter Aiken et al., *Microsoft Computer Dictionary*, (5th ed., Microsoft Press, 2002). Therefore, the word "processing" directly indicates a transformation of data. Furthermore, the term "application" is defined as "[a] program designed to assist in the performance of a specific task, such as word processing, accounting or inventory management." (Emphasis added) Peter Aiken et al., *Microsoft Computer Dictionary*, (5th ed., Microsoft Press, 2002). Therefore, the term "application," by definition, directly indicates a well-established utility (i.e., a practical application). Therefore, it is submitted that amended Claims 1-9 and 11-14 are not directed to abstract ideas and clearly comply with the requirements of 35 U.S.C. § 101. Consequently, applicants respectfully request that this ground of rejection be withdrawn with respect to these claims rejections stated in the Office Action.

LAW OFFICES OF  
CHRISTENSEN O'CONNOR JOHNSON KINDNESS<sup>PLLC</sup>  
1420 Fifth Avenue  
Suite 2800  
Seattle, Washington 98101  
206.682.8100

#### Claims 10 and 15-34

The Office Action rejected the above-mentioned claims because these claims "are not limited to tangible embodiments." Applicants respectfully disagree with the basis of rejection of these claims. However, in the interest of expediency, applicants have amended Claims 10, 15, 16, and 23 and, thus, all the claims depending therefrom by adding the term "storage." Applicants submit that this addition makes it clear that the claims are directed to tangible embodiments, thereby rendering the 35 U.S.C. § 101 rejection moot.

#### Claims 16-34

The Office Action rejected the above-mentioned claims because "Claims 16-34 appear to be data structure per se that is nonfunctional." Amended independent Claim 16 recites, in its entirety:

16. A computer-readable storage medium having stored thereon a data structure, the data structure comprising:

a first data field containing data representing a global initialize function;

a second data field containing data representing a global terminate function; and

a third data field containing data representing an application function.  
(Emphasis added.)

"[I]f a machine is programmed in a certain new and unobvious way, it is physically different from the machine without that program; its memory elements are differently arranged. . . More than mere abstractions, the data structures are specific electrical or magnetic structural elements in a memory." (Emphasis added.) *In re Lowry*, 32 USPQ2d 1031, 1034, 1035 (Fed. Cir. 1994). Like *Lowry*, the present invention recites a data structure that is used to store useful data in the memory of a computer, thus making the computer physically different. It is unclear why the Office Action considers the data structure recited by Claims 16-34 to be

LAW OFFICES OF  
CHRISTENSEN O'CONNOR JOHNSON KINDNESS<sup>PLLC</sup>  
1420 Fifth Avenue  
Suite 2800  
Seattle, Washington 98101  
206.682.8100



"nonfunctional." It is well known in the industry that the function of a data structure is to hold data. As noted above, amended Claim 16 recites, *inter alia*, "a first data field containing data representing a global initialize function" (emphasis added), which clearly indicates that the recited data structure contains useful data representing a function. The recited data structure clearly performs the useful function of holding data for the useful purpose of representing a global initialize function. In a similar fashion, amended independent Claim 23 recites, *inter alia*, "the data structure comprising: a first data field containing data representing an application table, the application table comprising an application table entry" (emphasis added). The recited data structure clearly performs the useful function of holding data for the useful purpose of representing an application table. Therefore, it is submitted that the amended Claims 16 and 23, and all the remaining claims depending therefrom (i.e., Claims 17-22 and 24-34), are not directed to non-functional data structures and clearly comply with 35 U.S.C. § 101. Thus, applicants request that this ground of rejection be withdrawn with respect to the claims.

Rejection of Claims 1, 3, 5-8, 10, 11, 14-16, 18-28, and 30-33 under 35 U.S.C. § 102(e)

As noted above, Claims 1, 3, 5-8, 10, 11, 14-16, 18-28, and 30-33 were rejected under 35 U.S.C. § 102(e) by the Office Action for "being anticipated by Kossatchev." Each independent claim and the claims dependent therefrom, are addressed separately as follows.

Claims 23-28 and 30-33

Amended independent Claim 23 recites, in its entirety:

23. A computer-readable storage medium having stored thereon a data structure, the data structure comprising:

a first data field containing data representing an application table, the application table comprising an application table entry; and

LAW OFFICES OF  
CHRISTENSEN O'CONNOR JOHNSON KINDNESS<sup>PLLC</sup>  
1420 Fifth Avenue  
Suite 2800  
Seattle, Washington 98101  
206.682.8100

a second data field containing data representing a parameter table, the parameter table comprising a parameter table entry. (Emphasis added.)

Kossatchev does not teach or suggest a first data field containing data representing an application table and a second data field containing data representing a parameter table. Kossatchev discloses a means 12 for generating specifications of the procedure interface 4 and a test source generator 14 that generates test source code based on the specifications. The specifications and test sources are stored in a repository 16. Kossatchev, Col. 3, lines 6-13. The repository 16 is not even used to store a table, but the specifications and test sources, let alone storing a first and a second data field containing data representing an application table and a parameter table, respectively. Therefore, it is respectfully submitted that amended Claim 23 is allowable.

Claim 24 depends from Claim 23 and is submitted to be allowable for at least the same reasons as presented for Claim 23. Additionally, Claim 24 recites "a third data field containing data representing a global initialize function; a fourth data field containing data representing a global terminate function; and a fifth data field containing data representing an application function." (Emphasis added.) Kossatchev does not teach or suggest a data structure comprising a third data field containing data representing a global initialize function. Kossatchev discloses a test driver generator 30 that provides functions to initialize the procedure interface 4, prepare input values, call tested procedures with test case parameters, and receive test results. Kossatchev, Col. 4, lines 31-36. Kossatchev does not teach or suggest that the test driver is represented by data contained in a data field of a data structure in any form. Furthermore, Kossatchev does not teach or suggest that the functions provided to initialize the procedure interface 4 are global. Kossatchev discloses that the test driver provides functions that are limited to initialize a specific procedure interface 4. In contrast to Kossatchev, Claim 24 recites a data field containing data representing a global initialize function. Kossatchev does not teach

or suggest a fourth data field containing data representing a global terminate function. Kossatchev discloses a test plan that "is a program that defines the order of script driver calls. "The test plan also checks the script driver call conditions and termination correctness." (Emphasis added.) Kossatchev, Col. 15, lines 44-47. Kossatchev does not teach or suggest the use of a global terminate function represented by a fourth data field in a data structure. Kossatchev discloses a program (i.e., the test plan) that checks the correctness of termination of a procedure, which was called by the test driver and subsequently terminated after execution. Therefore, Claim 24 is further submitted to be allowable for the additional reasons discussed above.

Claim 25 depends from Claim 24 and is submitted to be allowable for at least the same reasons discussed above with respect to Claim 24. Additionally, Claim 25 recites "a sixth data field containing data representing an application test function." (Emphasis added.) Kossatchev does not teach or suggest an application test function represented by data in a data field. Kossatchev discloses specifying the "behavior of a group of parallel procedures to be tested in a parallel mode separately from consecutive procedures." Kossatchev, Col. 1, lines 44-46. Therefore, Claim 25 is further submitted to be allowable for the reasons additional discussed above.

Claim 26 depends from Claim 24 and is submitted to be allowable for at least the same reasons discussed above with respect to Claim 24.

Claim 27 depends from Claim 26 and is submitted to be allowable for at least the same reasons discussed above with respect to Claim 26. Additionally, Claim 27 recites "a seventh data field containing data representing an application post function." (Emphasis added.) Kossatchev does not teach or suggest an application post function represented by data in a data field. Kossatchev discloses a test suite 22 that executes tests on a system under test 3 and

analyzes results of the tests to verify the procedure interface 4. Kossatchev, Col. 3, lines 60-63. Therefore, Claim 27 is further submitted to be allowable for the additional reasons discussed above.

Claim 28 depends from Claim 27 and is allowable for at least the same reasons discussed above with respect to Claim 27. Additionally, Claim 28 recites "an eighth data field containing data representing an application post test function." (Emphasis added). Kossatchev does not teach or suggest an application post test function represented by data in a data field. Kossatchev discloses a test driver that checks the "correctness of the target procedure execution results." Kossatchev, Col. 4, lines 55-60. Therefore, Claim 28 is further submitted to be allowable for the additional reasons discussed above.

Claim 30 depends from Claim 23 and is submitted to be allowable for at least the same reasons discussed above with respect to Claim 23. Additionally, Claim 30 recites "a third data field containing data representing a name of a parameter; a fourth data field containing data representing a type of the parameter; and a fifth data field containing data representing a value of the parameter." (Emphasis added). Kossatchev does not teach or suggest a name of the parameter, a type of the parameter, and a value of the parameter represented by data in a data field. Kossatchev discloses that "the test case parameter generator 32 generates constant arrays and programs that generate and select needed test case parameters." (Emphasis added.) Kossatchev, Col. 4, lines 13-16. Kossatchev further discloses that "[a] test case is an instance of a tested procedure. A test case is defined by a procedure name and its parameters. ...The test drivers use the test case parameters and execute test cases on the SUT 3 to verify the procedure interface 4." (Emphasis added.) Kossatchev, Col. 4, lines 25-29. Kossatchev discloses the use of parameters as part of the definition of a test case, in contrast to Claim 30, which recites name,

type, and value of a parameter represented by data in a data field. Therefore, Claim 30 is further submitted to be allowable for the additional reasons discussed above.

Claims 31, 32, and 33 depend from Claim 23 and are submitted to be allowable for at least the same reasons discussed above with respect to Claim 23.

Claims 16, 18-21

Amended independent Claim 16 recites, in its entirety:

16. A computer-readable storage medium having stored thereon a data structure, the data structure comprising:

a first data field containing data representing a global initialize function;

a second data field containing data representing a global terminate function; and

a third data field containing data representing an application function.  
(Emphasis added.)

Claim 16 is submitted to be allowable for at least the same reasons discussed above with respect to Claim 24.

Claims 18-21 depend from Claim 16 and are submitted to be allowable for at least the same reasons discussed above with respect to Claim 16. Additionally, Claims 18-21 are submitted to be allowable for at least the same additional reasons discussed above with respect to Claims 25-28.

Claims 1, 3, 5-8, and 10

Claim 1 recites, in its entirety:

1. A method for using a framework module to run an application, the framework module comprising an application table and a parameter table, the application table comprising one or more application table entries, the parameter table comprising one or more parameter table entries, the method comprising:

selecting an application table entry; and

LAW OFFICES OF  
CHRISTENSEN O'CONNOR JOHNSON KINDNESS<sup>PLLC</sup>  
1420 Fifth Avenue  
Suite 2800  
Seattle, Washington 98101  
206.682.8100

processing the selected application table entry, the processing comprising:

running a global initialize function referenced by the selected application table entry, running a sub-application referenced by the selected application table entry with one or more parameters referenced by one or more parameter table entries, and running a global terminate function referenced by the selected application table entry. (Emphasis added.)

Claim 1 is submitted to be allowable for at least the same reasons discussed above with respect to Claims 16 and 23. Additionally, Kossatchev does not teach or suggest a framework module to run an application, where the framework comprises an application table and a parameter table. Kossatchev discloses specifying the "behavior" of a group of parallel procedures to be tested in a parallel mode separately from consecutive procedures." Kossatchev, Col. 1, lines 44-46. (Emphasis added.) Furthermore, Kossatchev does not teach or suggest running a global initialize function referenced by the selected application table entry. Kossatchev also does not teach or suggest running a sub-application referenced by the selected application table entry with parameters referenced by parameter table entries. Kossatchev discloses, in Figure 2, a means 12 for generating specifications of the procedure interface 4 and a test source generator 14, which generates test source code based on the specifications. The specifications and test sources are stored in a repository 16. Kossatchev, Col. 3, lines 6-13. This is in contrast to Claim 1, which recites the use of an application table and a parameter table containing references to sub-applications and corresponding parameters, respectively, to run the sub-applications. Therefore, Claim 1 is further submitted to be allowable for the additional reasons discussed above.

Claim 3 depends from Claim 1 and is submitted to be allowable for at least the same reasons discussed above with respect to Claim 1.

Claims 5-8 depend from Claim 1 and are submitted to be allowable for at least the same reasons discussed above with respect to Claim 1.

LAW OFFICES OF  
CHRISTENSEN O'CONNOR JOHNSON KINDNESS<sup>PLLC</sup>  
1420 Fifth Avenue  
Suite 2800  
Seattle, Washington 98101  
206.682.8100

Claim 10 depends from Claim 1 and is submitted to be allowable for at least the same reasons discussed above with respect to Claim 1.

Claims 11 and 14-15

Independent Claim 11 recites, in its entirety:

11. A method for building a framework module for running an application, the framework module comprising an application table and a parameter table, the application comprising one or more sub-applications, the method comprising:

collecting data specifying one or more sub-applications composing the application;

collecting data specifying one or more parameters to the one or more sub-applications;

creating the application table, the creating of the application table comprising creating an application table entry for each of the one or more specified sub-applications, the creating of an application table entry comprising creating a reference to a global initialize function, creating a reference to a global terminate function, and creating a reference to the sub-application; and

creating the parameter table, the creating of the parameter table comprising creating a parameter table entry for each of the one or more specified sub-application parameters, the creating of a parameter table entry comprising creating a reference to a name of the parameter and creating a reference to a type of the parameter. (Emphasis added.)

Claim 11 is submitted to be allowable for at least the same reasons discussed above with respect to Claims 1, 16, and 23.

Claims 14 and 15 depend from Claim 11 and are submitted to be allowable for at least the same reasons discussed above with respect to Claim 11.

LAW OFFICES OF  
CHRISTENSEN O'CONNOR JOHNSON KINDNESS<sup>PLLC</sup>  
1420 Fifth Avenue  
Suite 2800  
Seattle, Washington 98101  
206.682.8100

Rejection of Claims 2, 4, 9, 12, 13, 17, 29, and 34 under 35 U.S.C. § 103(a)

As noted above, Claims 2, 4, 9, 12, 13, 17, 29, and 34 were rejected under 35 U.S.C. § 103(a) by the Office Action for "as being unpatentable over Kossatchev...in view of Grey." Each set of claims as grouped by the Office Action on pages 8 and 9 are addressed separately as follows.

Claims 2, 12, and 17

Claim 2 depends from Claim 1 and is submitted to be allowable for at least the same reasons discussed above with respect to Claim 1. As noted above with respect to Claim 1, Kossatchev does not teach or suggest a global initialization function referenced by the selected application table entry. Grey fails to supply the teachings missing from Kossatchev. Grey discloses that "an unnamed synchronization object may be created that the user can access through an ActiveX reference variable." (Emphasis added.) Grey, Col. 23, lines 24-26. Grey further discloses that "user can use the ActiveX reference to the object in place of its name when performing operations on the Object...without performing a Create operation in each thread." Grey, Col. 23, lines 62-64. ActiveX reference variable is used by Grey to access a synchronization object, in contrast to Claims 1 and 2, which recite the use of an application table entry to run a global initialize function. Additionally, Claim 2 recites "at least one of the global initialize and the global terminate functions is a NULL function." (Emphasis added). In contrast, Grey discloses that "if the user specifies an empty string as the name for a synchronization object, then an unnamed synchronization object may be created that the user can access though an ActiveX reference variable." Grey, Col. 23, lines 23-26. Claim 2 recites a NULL function, in contrast to Grey, which discloses an empty string as the name for a synchronization object, which is different from a function. Therefore, Claim 2 is further submitted to be allowable for the additional reasons discussed above.

LAW OFFICES OF  
CHRISTENSEN O'CONNOR JOHNSON KINDNESS<sup>PLLC</sup>  
1420 Fifth Avenue  
Suite 2800  
Seattle, Washington 98101  
206.682.8100



Claims 12 and 17 are submitted to be allowable for at least the same reasons discussed above with respect to Claim 2.

Claims 4, 13, 22, and 29

Claim 4 depends from Claim 1 and is submitted to be allowable for at least the same reasons discussed above with respect to Claim 1. As noted above with respect to Claim 2, Kossatchev does not teach or suggest a global initialization function referenced by the selected application table entry. Grey fails to supply the teachings missing from Kossatchev. Grey discloses that "an unnamed synchronization object may be created that the user can access through an ActiveX reference variable." (Emphasis added.) Grey, Col. 3, lines 24-26. Claim 4 recites, *inter alia*, "running a thread initialize function referenced by the selected application table entry." (Emphasis added.) Grey discloses a method that is used during the development of a program using an application development environment. During the development of the application, the developer may specify a batch synchronization section. Grey, Col. 12, lines 21-29. This is in contrast to Claim 2, which recites running a function referenced by the selected application table entry in a run-time environment. Therefore, Claim 4 is further submitted to be allowable for the additional reasons discussed above.

Claim 13 depends from Claim 11 and is submitted to be allowable for at least the same reasons discussed above with respect to Claim 11. As noted above with respect to Claim 11, Kossatchev does not teach or suggest running a global initialize function referenced by the selected application table entry. Kossatchev also does not teach or suggest running a sub-application referenced by the selected application table entry with parameters referenced by parameter table entries. Grey fails to supply the teachings missing from Kossatchev. Grey discloses a method whereby each thread in the plurality of threads may execute the program until the thread arrives at the enter point for the batch synchronization section. The thread is

blocked upon arriving at the enter point until all other threads arrive at the enter point. Once all threads have arrived at the enter point, execution of the program within the batch synchronization section proceeds. Grey, Col. 5, lines 52-58. Grey discloses a method for creating a computer program using batch synchronization section, whereas Claim 13 recites creating a reference to a thread initialize function and thread terminate function. Therefore, Claim 13 is further submitted to be allowable for the additional reasons discussed above.

Claim 22 recites features similar to Claim 13 and is, therefore, submitted to be allowable for at least the same reasons discussed above with respect to Claim 13.

Claim 29 recites features similar to Claim 13 and is, therefore, submitted to be allowable for at least the same reasons discussed above with respect to Claim 13.

Claim 9 depends from Claim 1 through Claim 8 and is submitted to be allowable for at least the same reasons discussed above with respect to Claim 2, which also depends from Claim 1.

Claim 34 depends from Claim 23 and is submitted to be allowable for at least the same reasons discussed above with respect to Claim 29, which also depends from Claim 23.

### CONCLUSION

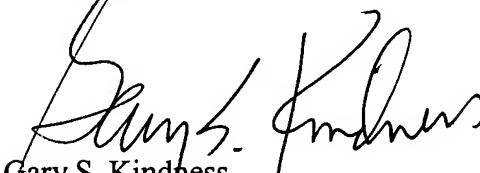
In summary, applicants respectfully submit that all the claims in this application comply with 35 U.S.C. § 101 and are clearly allowable in view of the disclosures of Kossatchev and Grey, applied singly or in any motivated combination. As a result, applicants respectfully request that all of the claims remaining in this application be allowed and this application be

LAW OFFICES OF  
CHRISTENSEN O'CONNOR JOHNSON KINDNESS<sup>PLLC</sup>  
1420 Fifth Avenue  
Suite 2800  
Seattle, Washington 98101  
206.682.8100

passed to issue. If the Examiner has any question, the Examiner is invited to contact applicants' attorney at the number set forth below.


Respectfully submitted,

CHRISTENSEN O'CONNOR  
JOHNSON KINDNESS<sup>PLLC</sup>

  
Gary S. Kindness  
Registration No. 22,178  
Direct Dial No. 206.695.1702

I hereby certify that this correspondence is being deposited with the U.S. Postal Service in a sealed envelope as first class mail with postage thereon fully prepaid and addressed to Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the below date.

Date: September 23, 2005



GSK/FXM:aew

LAW OFFICES OF  
CHRISTENSEN O'CONNOR JOHNSON KINDNESS<sup>PLLC</sup>  
1420 Fifth Avenue  
Suite 2800  
Seattle, Washington 98101  
206.682.8100